



## HTML

<b>1</b>	<b><i>Basics of HTML</i></b> .....	<b>1</b>
1.1	Elements .....	1
1.2	Attributes.....	4

# 1 Basics of HTML

## 1.1 Elements

`<html>`:

What: The root element that encapsulates the entire HTML document.

Usage: It contains metadata and references to external resources like stylesheets and scripts.

Default Display: Block-level element (fills the entire width of its parent).

`<head>`:

What: Contains meta-information about the document, such as the page title, character set, and links to external resources.

Usage: Not displayed on the web page itself but contains essential document information.

Default Display: Block-level element.

`<title>`:

What: Sets the title of the web page, which appears in the browser's title bar or tab.

Usage: Important for search engine optimization (SEO) and user experience.

Default Display: Inline element (within a line of text).

`<meta>`:

What: Provides metadata about the HTML document, including character encoding, author information, and viewport settings.

Usage: Helps browsers and search engines understand document characteristics.

Default Display: Not applicable (metadata).

`<link>`:

What: Used to link external resources like stylesheets (CSS) to the HTML document.

Usage: Helps with styling and layout of the web page.



Default Display: Not applicable (metadata).

`<script>`:

What: Embeds or references JavaScript code, allowing for dynamic and interactive behavior on the web page.

Usage: Can be placed in the `<head>` or at the end of the `<body>`.

Default Display: Not applicable (script content).

`<body>`:

What: Contains the visible content of the web page, including text, images, links, and other HTML elements.

Usage: It's where the main content of the page resides.

Default Display: Block-level element.

`<h1>` to `<h6>`:

What: Headings used to structure and organize content hierarchically.

Usage: `<h1>` is the highest level (most important) heading, while `<h6>` is the lowest level.

Default Display: Block-level elements.

`<p>`:

What: Represents a paragraph of text.

Usage: Used to separate and structure textual content on the page.

Default Display: Block-level element.

`<a>`:

What: Defines hyperlinks (anchors) that link to other web pages or resources.

Usage: Allows users to navigate to different parts of the web or external websites.

Default Display: Inline element.

`<ul>`:

What: Represents an unordered list of items.

Usage: Typically used for creating bulleted lists.

Default Display: Block-level element.

`<ol>`:



What: Represents an ordered list of items.

Usage: Used for creating numbered lists.

Default Display: Block-level element.

`<li>`:

What: Defines an individual item within an unordered or ordered list.

Usage: Contains list items.

Default Display: Block-level element.

`<img>`:

What: Embeds images on the web page.

Usage: To display images and visual content.

Default Display: Inline element.

`<div>`:

What: A generic container element used for layout and structuring content.

Usage: Provides a way to group and style elements together.

Default Display: Block-level element.

`<span>`:

What: A generic inline container element used for styling and grouping inline content.

Usage: Often used to apply styles or scripting to specific text within a larger block of text.

Default Display: Inline element.

`<table>`:

What: Defines a table for tabular data.

Usage: Used to organize and display data in rows and columns.

Default Display: Table element.

`<tr>`:

What: Represents a table row within a `<table>`.

Usage: Contains table data cells (`<td>`) or table header cells (`<th>`).

Default Display: Table row element.



`<td>`:

What: Represents a table data cell within a `<tr>`.

Usage: Contains data or content within a table row.

Default Display: Table cell element.

`<form>`:

What: Defines an HTML form for user input.

Usage: Used to collect user data and submit it to a server.

Default Display: Block-level element.

Common `<form>` attributes:

1. **action:** Specifies the URL to which the form data should be sent upon submission.
2. **autocomplete:** Enables or disables the browser's autofill feature for form fields.
3. **enctype:** Specifies the encoding type for form data when using "POST" method, usually "application/x-www-form-urlencoded" or "multipart/form-data."
4. **id:** Assigns a unique identifier to the form for JavaScript interactions and CSS styling.
5. **method:** Defines the HTTP method to be used for form submission, typically "GET" or "POST."
6. **name:** Provides a name for the form element, often used for identification in server-side processing.
7. **novalidate:** Prevents browser form validation, allowing custom JavaScript validation.
8. **onreset:** Defines JavaScript code to execute when the form is reset.
9. **onsubmit:** Specifies JavaScript code to execute when the form is submitted.
10. **target:** Determines where the form response should be displayed, typically "\_blank" for a new window or "\_self" for the current window.

## 1.2 Attributes

1. **action:** Specifies the URL to which the form data should be sent upon submission.
2. **alt:** Alternative text for images.
3. **alt:** Alternative text for screen readers.
4. **async:** Loads scripts asynchronously.



5. autocomplete: Enables/disables autofill.
6. autofocus: Auto-focuses input fields.
7. autoplay: Auto-plays media.
8. charset: Character encoding.
9. checked: Preselects checkboxes/radio buttons.
10. class: Assigns a class to an element for styling.
11. cols: Defines textarea columns.
12. controls: Displays media controls.
13. data-\*: Custom data attributes.
14. defer: Delays script execution.
15. disabled: Disables an element.
16. download: Specifies a downloadable link.
17. for: Associates a label with an input.
18. height: Sets the element's height.
19. hidden: Hides elements from display.
20. href: Defines the link destination.
21. id: Unique identifier for an element.
22. lang: Language for element content.
23. loop: Repeats media playback.
24. max: Maximum value for input fields.
25. media: Media query for styles.
26. method: Defines the HTTP method to be used for form submission, typically "GET" or "POST."
27. min: Minimum value for input fields.
28. name: Provides a name for the form element, often used for identification in server-side processing.
29. nowrap: Prevents text wrapping.
30. onchange: Executes JavaScript on change.
31. onclick: Executes JavaScript on click.
32. onload: Executes JavaScript on page load.
33. onsubmit: Executes JavaScript on form submission.



- 34. pattern: Validates input using a regex.
- 35. placeholder: Placeholder text for form input.
- 36. preload: Specifies media loading behavior.
- 37. readonly: Makes input fields read-only.
- 38. rel: Relationship between the current and linked document.
- 39. required: Indicates a required field.
- 40. rows: Defines textarea rows.
- 41. selected: Preselects options in a dropdown.
- 42. src: Specifies the source URL, often for images.
- 43. step: Increment value for number inputs.
- 44. style: Inline CSS styles.
- 45. target: Opens links in a new window/tab.
- 46. target: Determines where the form response should be displayed, typically "\_blank" for a new window or "\_self" for the current window.
- 47. title: Provides additional information.
- 48. type: Specifies content type.
- 49. value: Value for form elements.
- 50. width: Sets the element's width.